ADVERSIAL ANALYSIS OF EVOLUTIONARY MODELS AND MULTI-AGENT SYSTEMS

(Toward theoretical foundations for generative social science)

G. ISTRATE,* Los Alamos National Laboratory, Los Alamos, NM

Generative Social Science

- Epstein (Complexity '99): "If you didn't grow it you didn't explain it !"
- Epstein (2005) "To explain a macroscopic regularity x is to furnish a suitable microspecification that suffices to furnish it".
- Similar concerns in evolutionary game theory.
- Classical game theory: steady-state. How do equilibria arise ?
- Evolutionary game theory: equilibria arise as a result of a "learning" process.

Stochastically stable states

- Best-reply learning dynamics can lead to multiple equilibria (path dependence).
- (Peyton Young) Adding small amounts of noise to best-reply dynamics can lead to **equilibrium selection**.
- Noise (small deviations from rationality): generative explanation for equilibrium selection

* *Corresponding author address*: Gabriel Istrate, CCS-5, Basic and Applied Simulation Science, Los Alamos National Laboratory, P.O. Box 1663, Mail Stop M997, Los Alamos, NM 87545; e-mail: istrate@lanl.gov

Games/simulations as dynamical systems

- Multiagent simulations: interacting, nondeterministic dynamical systems.
- Robustness concerns: specification of interaction network, scheduling, dynamics.
- Most models assume some form of random scheduling. Not really plausible. Scheduling can make a difference (Huberman and Glance). Theory ?
- Approach: increase robustness of the models by considering adversarial scheduling.

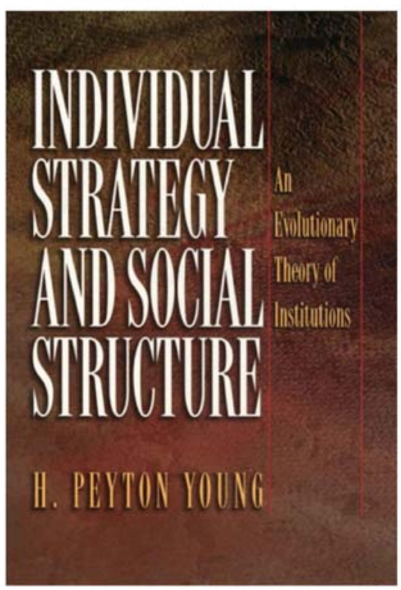
Approach (II)

- Start with base case result with random scheduling.
- Isolate properties of random scheduling.
- Gradually eliminate some of these properties ...
- ... Until base case result no longer true.
- Identify feature of scheduler responsible for the failure.
- Eliminate this property (result holds again), etc.
- In the course of this process: *add more realistic features, more robust restatement of results.*

Setup

- Population games (Blume): agents at the vertices of a graph. Each agent has a *state*.
- When agent scheduled, play a game against some of its neighbors. Changes state as a result of game playing.
- Scheduler: specifies what agent can get scheduled at what time.

Example I: emergence of institutions



Example I: emergence of institutions

strategies	A	B
A	a,a	c,d
B	d,c	b,b

- Strategy A is a *strict risk-dominant equilibrium*.
- That is $a - d > b - c > 0$.
- Selection of risk-dominant equilibria: Harsanyi and Selten.

Specifying dynamics

- When scheduled *agents play using the same strategy against each of their neighbors*.
- $\nu_i(z, \overline{x}_{-i})$, the payoff of the i'th agent, should he play strategy z with its neighbors is

$$\nu_i(z, \overline{x}_{-i}) = \sum_{(i,j) \in E} w_{i,j} a_{z,x_j}.$$

- If agent i is the one to update, $\overline{x}$ is the joint profile of agents' strategies, and $z \in \{\mathbf{A}, \mathbf{B}\}$ is the candidate new state,

$$p^{\beta}(x_i \rightarrow z | \overline{x}) \sim e^{\beta \cdot \nu_i(z, \overline{x}_{-i})},$$

Base case result

- Peyton Young: *under random scheduling the "all A" state the uniquely stochastically stable state.*
- Model of emergence of standards: gold vs. silver, driving on the left vs. right.
- Unrealistic feature: random norm adoption. No account of norm diffusion.

Properties of random schedulers

A random scheduler is:

(i) uniform: probability of getting scheduled is same.

(ii) non-adaptive: who gets scheduled does not depend on the past.

(a) who gets scheduled does not depend on *who got scheduled in the past.*

(b) who gets scheduled does not depend on the *past outcome of game-playing.*

(iii) fair: in $\theta(n \log n)$ steps all nodes get scheduled with probability 1.

Adversarial analysis

- *allow nonuniformity (drop (i)):* similar result to the one for baseline case.
- *allow adaptiveness (drop (ii a+b)):* can be just as fair as random scheduler and prevent stabilization.
- *only drop ii (b):* assume "social network of influences" (not necessarily the same as the game playing one). Scheduler: *random walk on this network*. Result again similar to the one for random scheduler.
- by now what was easy to show for random scheduling is quite nontrivial mathematically.

Making the result more robust

- *Time* until convention emerges: important !
- *Peyton-Young (based on Morris)* . Provably *small-world like structure implies $\theta(n)$ convergence time* for random scheduling.
- *Not true* for model with contagion.
- Instead of $\theta(n)$: new graph parameter related to *hitting time*.
- The time component of Peyton-Young's result: now true for new parameter.

Example II: PD with Pavlov dynamics

- n agents, situated at the nodes of a graph G.
- Each agent has a label from the set $\{0, 1\}$.
- At time zero the labels are chosen either uniformly at random, or according a fixed (but otherwise arbitrary) global configuration.
- At each step two of the players, i, j, that are connected by an edge update their labels from $X(i)$, $X(j)$ to $X(i) + X(j)$ (mod 2).

Base case result

- $(0, 0) \rightarrow (0, 0)$
- $(1, 1) \rightarrow (0, 0)$
- $(0, 1) \rightarrow (1, 1)$
- random scheduling: "all zero" unique fixed point, reached with probability one *red for all graphs with no isolated vertices*
- Convergence time (G. et al. 2002): exponential on complete graph, star graphs, $O(n \log n)$ on a cycle.
- Nonreversible Markov chain. Correlation: network structure $\rightarrow$ convergence time really nontrivial. More results (Mossel and Roch, arxiv.org/math.PR October 2005)

Genealogy of the model

- Shoham and Tennenholtz (AIJ 1994) "Colearning", distributed coordination model.
- Kittock (SFI proceedings 1994) experiments, this dynamics.
- Axelrod: Pavlov dynamics for IPD.
- Sidowsky "minimal social situation", Thibaut and Kelley, "mutual fate control" (1959).
- Coleman n player MSS (2005).

Types of scheduler and issues

- an edge-daemon is able to choose *both* players of the interacting pair.
- node-daemons choose only one of the players. The other player: random among the neighbors.
- *fairness*
- *adaptiveness*

Adversarial scheduling: results

- *Edge daemons are too strong*. One can preclude stabilization on "almost all" graphs, even for a non-adaptive daemon.
- *Nonadaptive node daemons:* similar to random schedulers.
- *Adaptive node daemons:* similar to random schedulers on almost all graph topologies (in random graph sense).

Convergence time

No mathematical results for convergence times adversarial scheduling. Convergence time seems consistent with the $O(n \log n)$ convergence time for random schedulers.

$\pi \mid n$	4	8	16	32	64	128
id	2.486	4.225	6.401	8.33	10.498	13.135
p3	2.469	4.039	5.807	7.662	9.639	11.718
$\pi \mid n$	256	512	1024			
id	16.091	17.954	20.331			
p3	14.323	16.054	19.826			

What about simulations ?

- other mathematical model (omitted) Schelling's segregation model (Peyton Young).
- model checking: technique used for hardware verification. Search for "bad events".
- scheduler: automaton. "Bad event": formula in temporal logic. Techniques from automata theory (Vardi and Wolper).
- More robust: model checking for interactive Markov chains (Herrmans).
- **LONG TERM:** adapting model checking MC to agent systems.

Conclusions

- Adversarial analysis is surprisingly feasible ...
- ... leads to robust results ...
- ... and could be used for agent-based simulations as well.

Theoretical results:

- with M.V. Marathe (VBI), S.S. Ravi (SUNY Albany CS).
- submissions to Games and Economic Behavior, Theoretical Computer Science. Available on request.